**LABORATORIO NO. 02**

“Ensamblador, DEBUG y Sistemas Numéricos”

|  |  |
| --- | --- |
| **Estudiante:** | **Carnet:** |
| LÓPEZ RAMÍREZ, ADOLFO BERNARDO | 1203612 |

**Ejercicio 1: Utilización del Ensamblador y el Enlazador**

Utilizando los archivos “Ejemplo1.asm” y “Ejemplo2.asm” genere el código objeto y el programa ejecutable utilizando el Ensamblador “TASM” y el Enlazador “TLINK”.

Conteste las siguientes preguntas:

1. **Cuando se genera el ejecutable del archivo “Ejemplo2.asm”, ¿cuál es la advertencia que se muestra en pantalla? ¿Por qué muestra esa advertencia?**

La advertencia que se genera es la siguiente: “Warning: no stack”. La cual nos indica que no se declaró el segmento de pila (o stack) dentro del programa que deseamos ejecutar.

1. **Modificado el programa para que no muestre la advertencia, ¿cuál es el resultado del programa, es decir, por qué se imprime ese carácter y no un 30?**

Se imprime el carácter \* debido a que nosotros estamos mandando a imprimir una suma de números hexadecimales por lo que el resultado no es el que nosotros deseamos. Tenemos 2 variables con valor de 15h y al sumarlo nosotros esperaríamos una respuesta de 30, sin embargo, el resultado de la suma es 2A, que es el valor hexadecimal para el carácter \* en ASSCII.

1. **Modifique el código del archivo “Ejemplo2.asm” y utilizando la tabla de códigos ASCII, imprima en pantalla una letra “Z”.**

**Ejercicio 2: Utilización del Modo “DEBUG”**

Comandos del Modo “DEBUG”:

* N Nombrar un programa.
* L Se encarga de cargar el programa.
* U        "Desensamblar" código máquina y pasarlo a código simbólico.
* A        Ensamblar instrucciones simbólicas y pasarlas a código máquina.
* D        Mostrar el contenido de un área de memoria.
* E        Introducir datos en memoria, iniciando en una localidad específica.
* G       Correr el programa ejecutable que se encuentra en memoria.
* P        Proceder o ejecutar un conjunto de instrucciones relacionadas.
* Q       Salir de la sesión con DEBUG.
* R       Mostrar el contenido de uno o más registros.
* T        Rastrear la ejecución de una instrucción.
* W       Escribir o grabar un programa en disco.

Utilizando el Modo “DEBUG” de DOS cargue el programa “Ejemplo2.exe” y responda las siguientes preguntas:

1. **¿En qué dirección de memoria inicia el código del programa?**

Empezó en la dirección de memoria 0B06:0100 F1

1. **¿En qué dirección de memoria termina el código del programa?**

Termina en la dirección de memoria: 0B06:011F 0A550A

1. **Aparecen los comentarios en pantalla ¿Sí? ¿No? ¿Por qué?**

Los comentarios no aparecen en la pantalla ya que en el modo debug solo aparecen las instrucciones no comentadas que se estén ejecutando en ese momento.

1. **Para cada una de las instrucciones del programa, escriba la dirección de memoria que tiene asignada:**

|  |  |
| --- | --- |
| **Dirección de memoria** | **Instrucción** |
| 0B06E:0000 B8700B | Mov AX,@DATA |
| 0B06E:0003 8ED8 | Mov DS,AX |
| 0B06E:0005 B80000 | Mov AX,0000h |
| 0B06E:0008 BB0000 | Mov BX,0000h |
| 0B06E:000B B015 | Mov AL,15h |
| 0B06E:000D B315 | Mov BL,15h |
| 0B06E:000F 02C3 | Add AL,BL |
| 0B06E:0011 B130 | Mov CL,30h |
| 0B06E:0013 02C1 | Add AL,CL |
| 0B06E:0015 8AD0 | Mov DL,AL |
| 0B06E:0017 B402 | Mov AH,02 |
| 0B06E:0019 CD21 | Int 21h |
| 0B06E:001B B44C | Mov AH,4CH |
| 0B06E:001D CD21 | int 21h |

1. **¿Cuál es la dirección del segmento de código?**

La dirección del segmento de código (CS) es: 0B6E

1. **Antes de iniciar la ejecución por pasos del programa, ¿cuáles son los valores de los registros de propósito general?**

|  |  |  |  |
| --- | --- | --- | --- |
| **AX** | **BX** | **CX** | **DX** |
| 0B70 | 0000 | 001F | 0000 |

1. **El valor del IP, ¿coincide con la dirección de inicio del programa?**

No, la dirección de inicio del programa empieza en 0B06E:0000 B8700B mientras que el valor IP es 0003.

1. **Utilice el comando para el rastreo instrucción por instrucción y, por cada línea del código, escriba el contenido de los registros internos del CPU.**





